

DRNN-based shift decision for automatic transmission

Advances in Mechanical Engineering
2020, Vol. 12(11) 1–8
© The Author(s) 2020
DOI: 10.1177/1687814020975291
journals.sagepub.com/home/ade

Kai-Qiang Ye¹ , Hong Gao¹, Ping Xiao^{1,2} and Pei-Cheng Shi¹

Abstract

In research on intelligent shift for automatic transmission, the neural network selected has no feedback and lacks an associative memory function. Thus, its adaptability needs to be improved. To achieve this, an automatic shift strategy based on a deep recurrent neural network (DRNN) is proposed. First, a neural network framework was designed in combination with an eight-speed gearbox that matches a particular type of vehicle. Then, the working principle of the DRNN was applied to the shifting process of an automatic gearbox, and the implementation model of the shift logic was established in MATLAB/Stateflow. A data sample obtained from the model was used to train the DRNN. Training and evaluation of the DRNN were accomplished in Python. Finally, a simulation comparison of the DRNN with a back-propagation (BP) neural network proved that after the epochs have been increased, the DRNN has higher precision and adaptation than a BP neural network. This research provides a theoretical basis and technical support for intelligent control of automatic transmission.

Keywords

Automatic transmission, automatic shift, DRNN

Date received: 12 December 2019; accepted: 24 June 2020

Handling Editor: James Baldwin

Introduction

For decades, the proportion of automatic transmission (AT) vehicles in the market has been increasing significantly because they are easy to operate, make drivers comfortable, and reduce driver fatigue.¹ Consequently, many alternative methods and theories have been proposed to solve the attendant problems. A key issue with conventional AT is low transfer efficiency. Many novel shift strategies have been proposed to improve driving comfort while taking vehicle dynamics and fuel economy into account.

Yi et al.² researched the control system of automatic vehicle transmission and used fuzzy control theory to achieve more reasonable shift occasion for the tracked vehicle. In 1992, Bavarian Motor Works of Germany developed an adaptive control system of four-speed and five-speed ATs that appropriately adjusts shift strategies by identifying the driver style and the environment and then shifting automatically. Deok-Hokim et al.³

proposed an adaptive neuro-fuzzy inference system combined with Lyapunov stability and robust adaptiveness for smooth and precise gear shifting.

Based on precedent, the driving environment,^{4,5} and the driver's intention,^{4,6} an advanced driver assistance system (ADAS)⁷ was further considered for avoiding or reducing unnecessary shift actions, and consequently, achieving better performance. Based on the shift strategy of fuzzy control, Kahlbau and Bestle^{8–10} researched how to avoid energy waste, and their simulation test demonstrated that their proposed shift

¹School of Mechanical and Automotive Engineering, Anhui Polytechnic University, Wuhu, China

²School of Engineering, University of Bridgeport, Bridgeport, CT, USA

Corresponding author:

Kai-Qiang Ye, School of Mechanical and Automotive Engineering, Anhui Polytechnic University, Beijing Middle Road, Wuhu City, 241000, China.
Email: 1239075922@qq.com



Creative Commons CC BY: This article is distributed under the terms of the Creative Commons Attribution 4.0 License (<https://creativecommons.org/licenses/by/4.0/>) which permits any use, reproduction and distribution of the work

without further permission provided the original work is attributed as specified on the SAGE and Open Access pages (<https://us.sagepub.com/en-us/nam/open-access-at-sage>).

strategy enhances fuel economy. Furthermore, some researchers optimized the shifting process of automatic transmission in other ways, such as minimizing the jerk¹¹ and reducing greenhouse gas emission.¹² Other researchers presented dynamic analyzes of mathematical models that were used to study the control strategy of hydromechanical continuously variable transmission (HMCVT)¹³ to improve shifting quality.

However, the control effect is related to design experience, for which it is hard to offer an optimal solution.¹⁴ The emergence of the neural network (NN) provides a novel solution to this contradiction. NN has a highly parallel structure and a capacity for parallel implementation that results in better fault tolerance and faster overall processing. The NN possesses an inherent nonlinear property that stems from its theoretical ability to approximate arbitrary nonlinear mapping. It brings a new promise to nonlinear control problems. A suitably trained NN has the ability to generalize all data. Thus, an NN can solve control processes that are hard to deal with using mathematical models or descriptive rules.¹⁵

Kondo and Goka et al.¹⁶ introduced the NN into shift strategy and set the road gradient, vehicle speed, braking force, and steering angle as input data, which made their model cover more extensive road conditions, particularly uphill and downhill conditions. However, more sensors are required to acquire several different types of data concurrently. The extra cost of placing these sensors in the vehicle would raise production costs, which would be unacceptable to both manufacturers and customers.¹⁷

Li et al.¹⁴ researched a high-speed, accurate NN optimization method for a shift schedule that meets real-time control requirements for construction vehicles in complex working environments. Li et al.¹⁷ proposed an adaptive downhill shift control system that considers vehicle state and driver intention in order to provide a reliable, effective solution to the downhill problem, and for comprehensive improvement of drivability, the strategy of uphill, sports, and heavy brake application.

However, although the NN selected in the above shift strategy is superior in comparison with a back-propagation NN (BPNN), it remains a feedforward NN with the implication that data are isolated from each other at different moments. DRNN is a kind of powerful artificial NN algorithm proposed based on the premise that *human cognition is based on past experience and memory*. DRNN is different from the traditional NN as it not only considers the input from the previous moment but gives the network a new function to *remember* the previous content. The name, deep recurrent neural network, implies that the current output of a sequence is also linked to the previous output.

Notably, the network will remember the previous information and apply it to the calculation of the current output, that is, the nodes between the hidden layers are no longer unconnected but connected, and the input of the hidden layer includes not only the output of the input layer but also the output of the hidden layer at the previous moment. Hence, it is particularly useful for processing sequential data such as sound, time series (sensor) data, or written natural language.

Because DRNNs have these advantages, the DRNN was applied to the gear shifting decision process in this study, and consequently, enhanced its use of data. In the subsequent sections of this paper, the principle of the DRNN is first introduced to explain how it works. Then, the automatic shift strategy based on DRNN is proposed, including data acquisition, preprocessing of training data, and training of DRNN. Finally, the BPNN gear shifting prediction model is set as the control group, and the simulation result shows that DRNN is indeed superior to BPNN in prediction accuracy. In the next phase, more data collected from varied vehicle models under complex conditions will train new neural network to adapt corresponding models and various road conditions. This research can provide technical guidance and theoretical basis for vehicle automation and intelligence.

Deep recurrent neural network

Structure of DRNN

The eight-speed gearbox matching a particular type of vehicle was selected as the research object. The relative fuel injection volume (ratio of the current fuel injection volume to the maximum fuel injection volume) and the speed were taken as the input of the DRNN. Thus, the format of the input layer is 2×1 . Then, the data of the current step and the previous two steps are set as the input data for each input. Because gear selection is a classification problem, the number of neuron nodes in the output layer is the type (label) of the output vector, which is also the number of gear positions. Thus, the number of neuron nodes in the output layer is eight. To improve the adaptability of the network and speed up the rate of convergence in training, a hidden layer with three recurrent layers is designed. Through experience and experimentation, we find that DRNN works best when the number of hidden layer nodes is 13.

RNN training algorithm: Back-propagation through time

Forward compute. The output of the recurrent layer s_t and the output of the output layer o_t at time step t are defined as:

$$o_t = g(Vs_t^i) \quad (1)$$

$$s_t^i = f(U^{i-1}s_t^{i-1} + W^i s_{t-1}^i) \quad (2)$$

$$\vdots \quad (3)$$

$$s_t^2 = f(U^1 s_t^1 + W^2 s_{t-1}^2)$$

$$s_t^1 = f(Gx_t + W^1 s_{t-1}^1) \quad (4)$$

Where $f(\cdot)$ is the activation function of the current layer, $G \in \mathbb{R}^{13 \times 2}$ is the weight of the input layer to the l th recurrent layer, $x_t \in \mathbb{R}^{2 \times 1}$ is the input, $W^i \in \mathbb{R}^{13 \times 13}$ is the weight of the i th recurrent layer at time step $t-1$ to the i th recurrent layer at time step t , $s_t^i \in \mathbb{R}^{13 \times 1}$ is the output of the i th recurrent layer at time steps t , $U^i \in \mathbb{R}^{13 \times 13}$ is the weight of the i th recurrent to the $i+1$ th recurrent layer, $g(\cdot)$ is the activation function of the output layer, and $V \in \mathbb{R}^{8 \times 13}$ is the weight of the last recurrent layer to the output layer.

Calculation of error terms. BP through time propagates the error of the layer h at time step t in two directions. In one of the directions, back-propagation through time (BPTT) transfers the error to the upper layer to obtain δ_t^{h-1} , which is related to only the weight matrix U . In the other direction, BPTT transfers it to the initial time step through time to obtain δ_1^h , which is related to only the weight matrix W .

The weighted input of the neuron at time step t is defined as:

$$net_t = Ux_t + Ws_{t-1} \quad (5)$$

Therefore, the output of the recurrent layer at time step $t-1$ is defined as:

$$s_{t-1} = f(net_{t-1}) \quad (6)$$

Partial derivatives of net_t with respect to net_{t-1} are defined as:

$$\frac{\partial net_t}{\partial net_{t-1}} = \frac{\partial net_t}{\partial s_{t-1}} \frac{\partial s_{t-1}}{\partial net_{t-1}} \quad (7)$$

$$\begin{aligned} \frac{\partial net_t}{\partial s_{t-1}} &= \begin{bmatrix} \frac{\partial net_t^1}{\partial s_{t-1}^1} & \cdots & \frac{\partial net_t^1}{\partial s_{t-1}^{13}} \\ \vdots & \ddots & \vdots \\ \frac{\partial net_t^{13}}{\partial s_{t-1}^1} & \cdots & \frac{\partial net_t^{13}}{\partial s_{t-1}^{13}} \end{bmatrix} \\ &= \begin{bmatrix} w_{1,1} & \cdots & w_{1,13} \\ \vdots & \ddots & \vdots \\ w_{13,1} & \cdots & w_{13,13} \end{bmatrix} = W \end{aligned} \quad (8)$$

$$\begin{aligned} \frac{\partial s_{t-1}}{\partial net_{t-1}} &= \begin{bmatrix} \frac{\partial s_{t-1}^1}{\partial net_{t-1}^1} & \cdots & \frac{\partial s_{t-1}^1}{\partial net_{t-1}^{13}} \\ \vdots & \ddots & \vdots \\ \frac{\partial s_{t-1}^{13}}{\partial net_{t-1}^1} & \cdots & \frac{\partial s_{t-1}^{13}}{\partial net_{t-1}^{13}} \end{bmatrix} \\ &= \begin{bmatrix} f'(net_{t-1}^1) & 0 & \cdots & 0 \\ 0 & f'(net_{t-1}^2) & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & f'(net_{t-1}^{13}) \end{bmatrix} \\ &= \text{diag}[f'(net_{t-1})] \end{aligned} \quad (9)$$

where $w_{j,i}$ is the weight from the neurons in recurrent layers at time step $t-1$ to the j th neurons in recurrent layers at time step t , s_{t-1}^k are the k th neurons in the recurrent layer at time step t , and net_{t-1}^j is the input of the j th neurons in the recurrent layer at time step t .

Thus, the final expression of equation (5) is defined as:

$$\frac{\partial net_t}{\partial net_{t-1}} = \frac{\partial net_t}{\partial s_{t-1}} \frac{\partial s_{t-1}}{\partial net_{t-1}} = W \text{diag}[f'(net_{t-1})] \quad (10)$$

This formula describes the rule that error terms propagate to the last moment through time. Based on this formula, an error term at any time step k is defined as:

$$\begin{aligned} \delta_k^T &= \frac{\partial E}{\partial net_k} = \frac{\partial E}{\partial net_t} \frac{\partial net_t}{\partial net_k} \\ &= \frac{\partial E}{\partial net_t} \frac{\partial net_t}{\partial net_{t-1}} \frac{\partial net_{t-1}}{\partial net_{t-2}} \cdots \frac{\partial net_{t-1}}{\partial net_k} \\ &= \text{Wdiag}[f'(net_{t-1})] \text{Wdiag}[f'(net_{t-2})] \cdots \text{Wdiag}[f'(net_k)] \delta_t^T \\ &= \delta_t^T \prod_{i=k}^{t-1} \text{Wdiag}[f'(net_i)] \end{aligned} \quad (11)$$

where δ_k^T is the matrix transpose of the error term at time step k . This formula is used to calculate the error term of the BP through time.

The relationship between the weighted inputs of the neighboring two cyclic layers are as follows:

$$net_t^l = Ua_{t-1}^{l-1} + Ws_{t-1} \quad (12)$$

$$a_{t-1}^{l-1} = f^{l-1}(net_{t-1}^{l-1}) \quad (13)$$

Where net_t^l is the weighted input of the l th layer at time step t , a_{t-1}^{l-1} is the output of the $l-1$ th layer at time step t , and f^{l-1} is the activate function of the $l-1$ th layer.

Thus, partial derivatives of net_t^l with respect to net_{t-1}^{l-1} are defined as:

$$\frac{\partial net_t^l}{\partial net_{t-1}^{l-1}} = \frac{\partial net_t^l}{\partial a_{t-1}^{l-1}} \frac{\partial a_{t-1}^{l-1}}{\partial net_{t-1}^{l-1}} = U \text{diag}[f'^{l-1}(net_{t-1}^{l-1})] \quad (14)$$

Hence, the matrix transpose of the error term of the previous layer is:

$$\begin{aligned}\delta_i^T &= \frac{\partial E}{\partial net_i^{l-1}} = \frac{\partial E}{\partial net_i^l} \frac{\partial net_i^l}{\partial net_i^{l-1}} \\ &= (\delta_i^l)^T Udiag[f'^{l-1}(net_i^{l-1})]\end{aligned}\quad (15)$$

This equation describes the rule that error terms propagate from one layer to the above layer. Based on this, the error terms of m th layer are defined as:

$$(\delta_i^m)^T = (\delta_i^l)^T \prod_{j=m}^{l-1} Udiag[f'^j(net_i^j)] \quad (16)$$

Weight gradient calculation. Because the partial derivatives of loss function E with respect to weight matrix W , which has a connection with only the present input of the recurrent layer has no connection with U and X_t , partial derivatives of loss function E with respect to each weight matrix W is defined as:

$$\frac{\partial E}{\partial w_{j,i}} = \frac{\partial E}{\partial net_j^t} \frac{\partial net_j^t}{\partial w_{j,i}} = \delta_j^t s_i^{t-1} \quad (17)$$

Weight modification. Because the stochastic gradient descent (SGD) has the advantage of automatic escape from the saddle point and local optimum, the SGD is chosen as the optimization algorithm.

The expression of weight modification is:

$$\hat{g} \leftarrow + \frac{1}{m} \nabla_w \sum_i E(f(x^{(i)}; W), y^{(i)}) \quad (18)$$

$$w \leftarrow w - \eta \hat{g} \quad (19)$$

where m is batch size, E is the loss function, $f(x^{(i)}; w)$ is the predicted output, $y(i)$ is the actual output, and η is the learning rate.

Automatic shift strategy based on DRNN

Prediction principle of shift

Considering the two-parameter shift situation for vehicle speed and relative fuel injection volume and for better consideration of vehicle dynamics and fuel economy, the realization flow chart of an intelligent shift program based on DRNN is provided in combination with the aforementioned theoretical model of the DRNN. The flowchart of the DRNN-based automatic shift strategy is presented in Figure 1.

In Figure 1, the Stateflow model of shift logic is first constructed using MATLAB to determine the critical speed and relative fuel injection volume of different shift positions. Then, the data between the shift critical value is filled completely as input data and expected

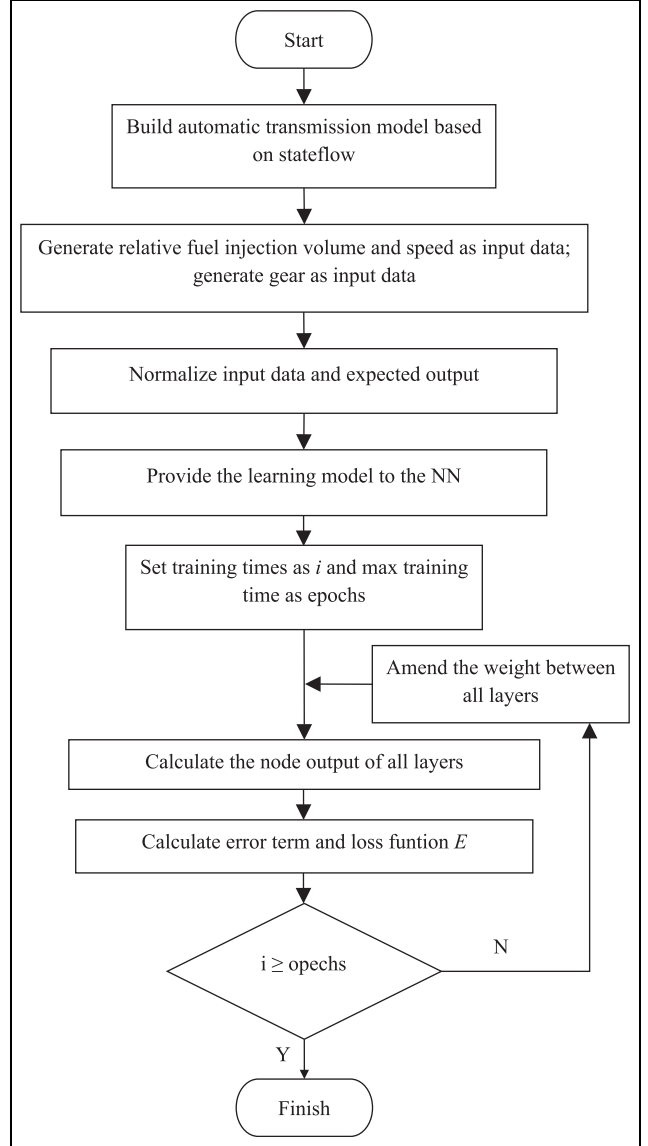


Figure 1. DRNN flowchart.

output. Because the dimensions of the input data are different from the dimensions of the expected output, and a DRNN is sensitive to the data, the data needs to be normalized as training samples. Subsequently, the number of neuron nodes of the input layer and the output layer are determined based on the dimensions of the input data and the actual output in the training sample, and the number of neuron nodes of the current layer is determined based on an empirical formula experiment. The structure of the NN is determined concurrently. After the NN is trained using the training data, the output and theoretical output of the recurrent layer is obtained. By comparing the actual output with the theoretical output and calculating their difference, we obtain the loss function and error term, which are used to modify the weight matrix between layers in the

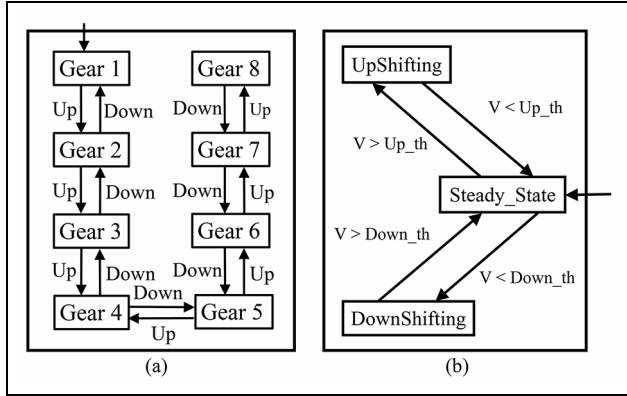


Figure 2. Stateflow model: (a) shift diagram and (b) shift logic diagram.

Table 1. Vehicles matching.

Basic parameters	Value
Power	230 horsepower
Engine speed	500–1990 r/min
First ratio	0.667
Second ratio	1
Third ratio	1.45
Fourth ratio	2
Fifth ratio	2.66
Sixth ratio	3.2
Seventh ratio	3.6
eighth ratio	4

NN. Finally, training of the NN is stopped when the epoch reaches the maximum. After the training is completed, this DRNN can be used to assist transmission with automatic shift control.

Establishment of shift logic Stateflow

The Stateflow model can realize gear selection for transmission, as shown in Figure 2. The input is defined as the relative fuel injection volume and vehicle speed by the Model Explorer, and the output is defined as the gear required. Selection-state (always active) starts by calculating the during function.

Based on the gear and the instantaneous value of the relative fuel injection volume, the velocity threshold of the gear is calculated. When the model is in steady-state, it compares these values with the current speed to determine whether to shift or not. If the speed does not meet the shift condition, the model will not shift and moves back to steady-state to avoid the shift caused by external noise, see Table 1.

Preprocessing of training data

DRNN is sensitive to data, and the dimensions of input data are different from the dimensions of expected

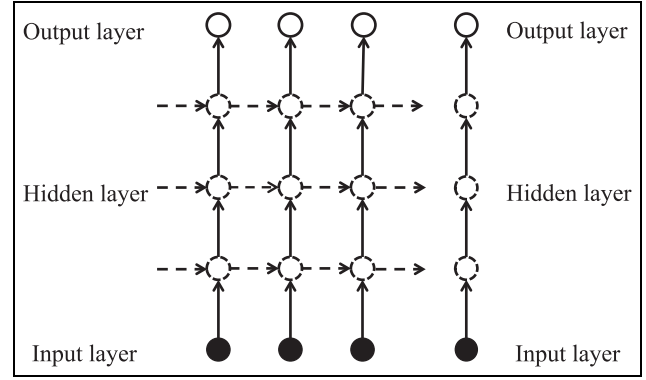


Figure 3. Structure diagram: (a) DRNN and (b) BPNN.

output. Hence, data needs to be normalized, and its normalization formula is defined as:

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \quad (20)$$

where x' is the normalized data, x is the data waiting to be normalized, x_{\min} is the smallest data in each column, and x_{\max} is the biggest data in each column.

Training

To highlight the superiority of DRNN in processing time series data,¹⁸ a BP NN (BPNN) is designed as the control group. This BPNN uses the hidden layer and output layer with the same number of neurons as the DRNN, and the output layer of the BPNN selects the same activation function and loss function as the DRNN. The structure diagram of the DRNN and BPNN are presented in Figure 3.

After the data was normalized, the required libraries were imported into Jupyter, and super parameters in the NN were set based on experience. When the data is inputted to both NNs, the weights in both NNs will be modified based on the algorithm introduced earlier, which makes both neural networks achieve the expected level of precision. Simultaneously, the image of loss varying with epochs was obtained and is presented in Figure 4. The loss of DRNN was eventually lower than 0.2; thus, achieving the expected accuracy.

Apparently, when compared with a DRNN, the BPNN has limitations dealing with complex problems, especially when dealing with time series problems.

Finally, the image of accuracy varying with epochs of the two neural networks is shown in the figure below. As shown in Figure 5, the accuracy of the BPNN tends to become flat after being trained 8000 times and finally fluctuates around 80%, which is not an ideal level of accuracy. It is easy to see that the BP neural network has some limitations in dealing with a two-parameter shift. After being trained, the accuracy of the DRNN

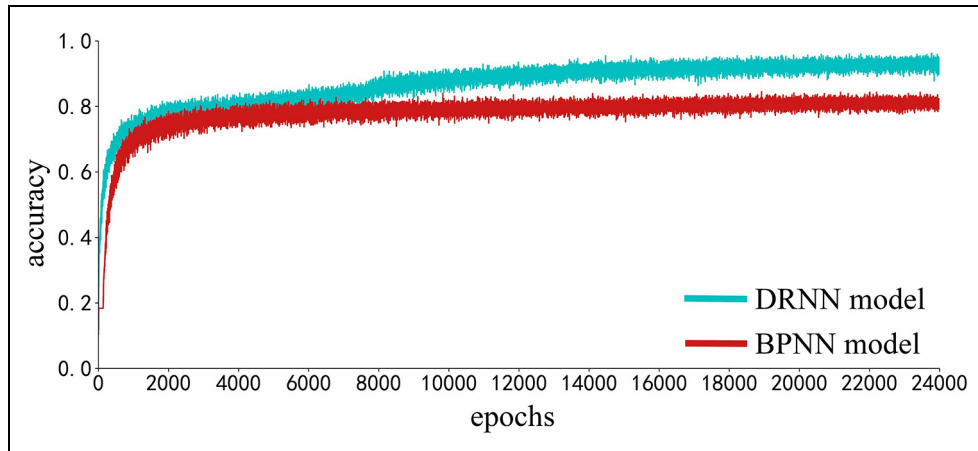


Figure 4. The image of accuracy varying with epochs.

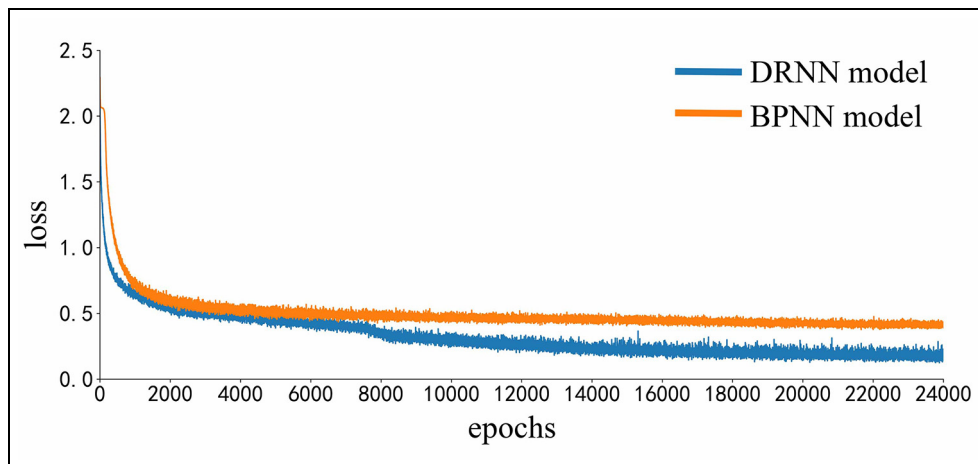


Figure 5. The image of loss varying with epochs.

Table 2. Comparison table of prediction accuracy between DRNN and BPNN.

The relative fuel injection	Acceleration		Deceleration	
	Low speed	High speed	Low speed	High speed
Small	–	+	++	+
Medium	0	++	++	–
Large	0	++	++	0

Note: “++” means that the prediction accuracy of DRNN is much higher than that of BPNN.

“+” means that the prediction accuracy of DRNN is slightly higher than that of BPNN.

“–” means that the prediction accuracy of DRNN is slightly lower than that of BPNN.

“0” means that the prediction accuracy of DRNN and BPNN are nearly equal.

will slowly increase, vary with epochs, and eventually reach 93%, which is far higher than the accuracy of the BPNN.

Simulation

After the training of the DRNN is completed, five groups of test data are selected as the input for the

DRNN and BPNN. Then, the prediction results for gear position can be obtained, as shown in Figure 6 and Table 2. Simulation results show that the DRNN has higher accuracy than the BPNN in predicting gear position. In addition, the prediction accuracy of DRNN is significantly higher than that of BPNN when the relative fuel injection volume is 0. Conversely, a comparison of Figure 6(b) and (c) shows that at the same

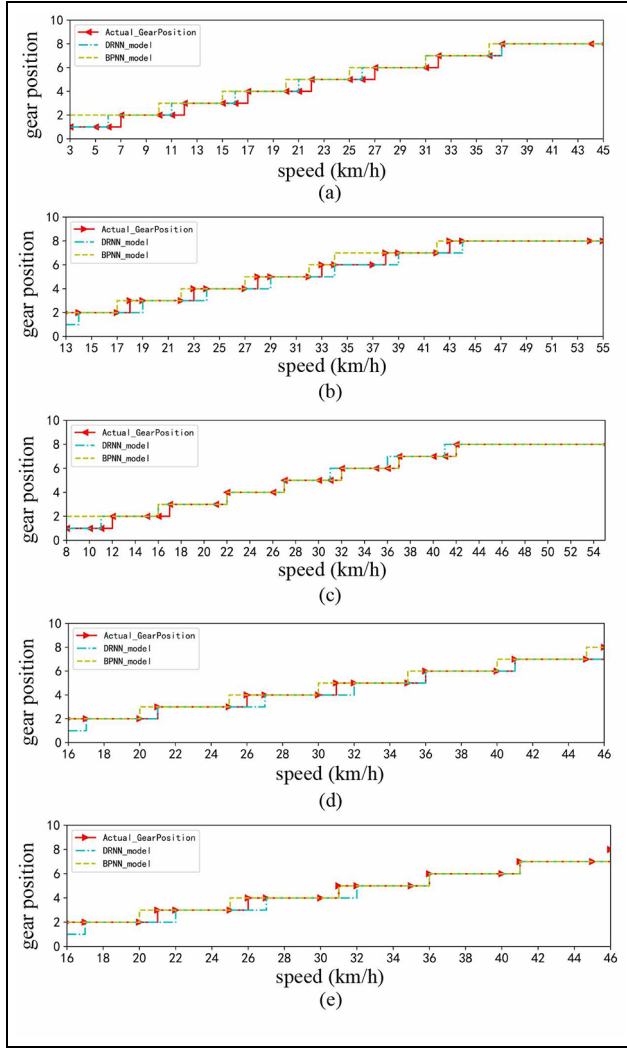


Figure 6. Comparison diagram of gear prediction: (a) deceleration; relative fuel injection volume = 0%, (b) acceleration; relative fuel injection volume = 50%, (c) deceleration; relative fuel injection volume = 50%, (d) acceleration; relative fuel injection volume = 90%, and (e) acceleration; relative fuel injection volume = 100%.

relative fuel injection volume and speed, the gear position in the downshift process is not lower than the gear position in the upshift process, which is in agreement with the test data provided. However, the BPNN, whether accelerating or slowing down, will give the same result when the input does not change. Finally, only through relative fuel injection volume and speed could the state of the vehicle (upshift or downshift) be judged, and the two parameters for high precision shift also be achieved.

Importantly, the Table 2 shows four things. First, the prediction accuracy of DRNN will increase with the decrease of the relative fuel injection volume when the vehicle is running at low speed. Second, the prediction accuracy of DRNN is slightly higher than that of

BPNN on the whole when the vehicle is running at high speed. Third, the DRNN model has more prediction accuracy for high-speed vehicles when the relative fuel injection volume is large. Last, the prediction accuracy of DRNN is significantly higher than that of BPNN when the vehicle is slowing down and in process of shift down.

Compared with previous research, the similarities between researches lies in that the excellent nonlinear fitting ability of neural network was sufficiently utilized. The difference lies in that a more advanced and more applicable neural network was applied, which further improved the prediction accuracy of neural network with its unique advantages in processing time series data. In addition, compared with other neural networks, DRNN can achieve the same function with fewer features.

Although there are important discoveries revealed by these studies, there are still shortcomings in this study. For example, the training data of vehicles under complex condition (such as mountains and muddy roads) are not obtained, which makes it difficult to guarantee the prediction accuracy of this model under complex condition.

As the current simulation results are relatively ideal, it has the prospect of being applied to other vehicles. If the data under complex working conditions can be obtained, the coverage of the model will certainly be wider.

This study does not apply the shifting model in a validated vehicle model with different driving cycle. Therefore, in the next phase of the study, this problem will be firstly solved and the data obtained from different vehicle under complex pavement will be used as extra training samples to train the new neural network to improve its adaptability and practicability.

Conclusion

For a transmission control system built with DRNN, we only need to obtain real-time data on relative fuel injection volume and speed to judge the vehicle's state (acceleration/deceleration) and determine the corresponding gear position that overcomes the shortcomings of the traditional two-parameter neural network shift.

By comparing the simulation results for the BPNN and DRNN, we find that the prediction accuracy of a DRNN reaches more than 93%, which is much higher than the 80% for the BPNN. This indicates that a DRNN has superior adaptability to that of a BPNN when applied to gear shifting in vehicles.

Acknowledgements

The authors would like to thank the financial support of the National Natural Science Foundation of China (Grant No.

51575001), the Major Natural Science Foundation of Anhui Provincial Universities (Grant No. KJ2017ZD14) and the Collaborative Innovation Project of Anhui Provincial University (Grant No. GXXT-2019-021) for their financial support.


Declaration of conflicting interests


The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.


Funding

The author(s) disclosed receipt of the following financial support for the research, authorship and/or publication of this article: The National Natural Science Foundation of China (Grant No.51575001), the Major Natural Science Foundation of Anhui Provincial Universities (Grant No. KJ2017ZD14) and the Collaborative Innovation Project of Anhui Provincial University (Grant No. GXXT-2019-021).

ORCID iDs

Kai-Qiang Ye  <https://orcid.org/0000-0002-8351-4466>

Ping Xiao  <https://orcid.org/0000-0002-2197-8804>

Pei-Cheng Shi  <https://orcid.org/0000-0003-1533-8154>

References

- Hui J and Anlin G. A new vehicle intelligent shift system. *Adv Mech Eng* 2015; 5: 303691–303691.
- Yi J, Wang XL, Hu YJ, et al. Modelling and simulation of a fuzzy controller of automatic transmission of a tracked vehicle in complicated driving conditions. *Proc IMechE Part D J Aut Eng* 2007; 221: 1259–1272.
- Kim DH, Yang KJ, Hong KS, et al. Smooth shift control of automatic transmissions using a robust adaptive scheme with intelligent supervision. *Int J Veh Des* 2003; 32: 250–272.
- Qin G, Ge A and Lee JJ. Knowledge-based gear-position decision. *IEEE Trans Intell Transp Syst* 2004; 5: 121–125.
- Guangqiang WU and Deming Z. Shifting rule modification strategy of automatic transmission based on driver-vehicle-road environment. *Chin J Mech Eng* 2010; 12: 346–352.
- Mashadi B, Kazemkhani A and Lakeh RB. An automatic gear-shifting strategy for manual transmissions. *Proc IMechE Pt I J Syst Contr Eng* 2007; 221: 757–768.
- Meng F and Jin H. Slope shift strategy for automatic transmission vehicles based on the road gradient. *Int J Automot Technol* 2018; 19: 509–521.
- Kahlbau S and Bestle D. Optimal shift control for automatic transmission#. *Mech Based Des Struct Mach* 2013; 41: 259–273.
- Wang J, Liu Y, Liu Q, et al. Power-based shift schedule for pure electric vehicle with a two-speed automatic transmission. *IOP Conf Ser: Mater Sci Eng* 2016; 157: 012016.
- Mashadi B and Lakeh RB. Fuel economy improvement based on a many-gear shifting strategy. *Int J Energy Environ* 2012; 3: 577–590.
- Haj-Fraj A and Pfeiffer F. Optimal control of gear shift operations in automatic transmissions. *J Franklin Inst* 2001; 338: 371–390.
- Newman K, Kargul J and Barba D. Development and testing of an automatic transmission shift schedule algorithm for vehicle simulation. *SAE Int J Eng* 2015; 8: 1417–1427.
- Zhu Z, Gao X, Pan D, et al. Study on the control strategy of shifting time involving multigroup clutches. *Math Prob Eng* 2016; 2016: 9523251.
- Li T, Liu H, Zhi-wen Z, et al. Shift scheduling strategy development for parallel hybrid construction vehicles. *J Cent South Univ* 2019; 26: 587–603.
- Cai ZX. *Intelligent control: principles, techniques and applications*. Singapore: World Scientific, 1997, p. 243.
- Kondo K and Goka H. Adaptive shift scheduling strategy introducing neural network in automatic transmission. *JSAE Rev* 1995; 16: 411–414.
- Li Y, Yu F and Wu C. Application of neural network to downhill shift strategy for automatic transmission. *J Shanghai Jiaotong Univ (Sci)* 2010; 15: 498–503.
- Goodfellow I, Bengio Y and Courville A. *Deep learning*. Cambridge: MIT Press, 2016, pp. 363–388.